



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/667,816	09/22/2003	John Phenix	14846-21	2173
7590	07/19/2010		EXAMINER	
Michael B. Johannessen, Esq. Lowenstein Sandler, P.C. 65 Livingston Avenue Roseland, NJ 07068				RADTKE, MARK A
ART UNIT		PAPER NUMBER		
		2165		
		MAIL DATE		DELIVERY MODE
		07/19/2010		PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte JOHN PHENIX

Appeal 2009-002724
Application 10/667,816
Technology Center 2100

Before JOHN A. JEFFERY, LEE E. BARRETT, and THU A. DANG,
Administrative Patent Judges.

DANG, *Administrative Patent Judge.*

DECISION ON APPEAL¹

¹ The two-month time period for filing an appeal or commencing a civil action, as recited in 37 C.F.R. § 1.304, or for filing a request for rehearing, as recited in 37 C.F.R. § 41.52, begins to run from the “MAIL DATE” (paper delivery mode) or the “NOTIFICATION DATE” (electronic delivery mode) shown on the PTOL-90A cover letter attached to this decision.

I. STATEMENT OF THE CASE

Appellant appeals from the Examiner's final rejection of claims 1, 3-5, and 7-9 under 35 U.S.C. § 134(a) (2002). We have jurisdiction under 35 U.S.C. § 6(b) (2002).

We reverse.

A. INVENTION

According to Appellant, the invention relates to the field of object-oriented software, and, more specifically, to a utility for differentiating between two objects and storing the differences for display (Spec. 1, ll. 12-14).

B. ILLUSTRATIVE CLAIM

Claim 1 is exemplary and is reproduced below:

1. A computer implemented method for comparing a first object and a second object in an object-oriented operating system comprising the steps of:

(a) determining whether the first object is equal to the second object; and if said objects are not equal:

(b) obtaining one or more methods from said first object and said second object;

(c) determining whether the one or more methods from said first object are equal to the one or more methods from said second object; and

(d) recursively performing steps (b) and (c) until all of the methods for the first object and the second object have been obtained;

- (e) generating a document comprising a listing of differences between the methods;
- (f) transforming the document into a human-readable form; and
- (g) displaying the human-readable form to a user.

C. REJECTIONS

The prior art relied upon by the Examiner in rejecting the claims on appeal is:

Keller	US 6,662,312 B1	Dec. 9, 2003
Mason	US 6,826,716 B2	Nov. 30, 2004

Shirazi, “Recursion and Stacks,” Java Performance Tuning § 7.5 (O’Reilly Media, Inc. 2000), *available at* <http://proquest.safaribooksonline.com/0-596-00015-4> (“Java”).

Claims 1, 3-5, and 7-9 stand rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter.

Claims 1, 3-5, and 7-9 stand rejected under 35 U.S.C. § 103(a) over the teachings of Mason in view of Java and Keller.

II. ISSUES

- 1) Has the Examiner erred in finding that claims 1, 3-5, and 7-9 are directed to non-statutory subject matter?
- 2) Has the Examiner erred in concluding that Mason in view of Java and Keller would have taught or suggested “determining whether the first object is equal to the second object” and “if said objects are not equal... obtaining one or more methods from said first object and said second object” as required by claim 1.

III. FINDINGS OF FACT

The following Findings of Fact (FF) are shown by a preponderance of the evidence.

Mason

1. Mason discloses testing Enterprise Web Applications to verify their correctness, reliability, availability, and security (col. 1, ll. 12-17).
2. The method includes finding 422 in a deployment descriptor for the Enterprise JavaBean (EJB) module 404 one or more <enterprise-bean> elements 412 having nested <ejb-name> sub-elements 414 having values equal to the first <ejb-name> 406 sub-element value (col. 11, ll. 30-34; Fig. 4).

VI. ANALYSIS

35 U.S.C. § 101

Though the Examiner finds that “[t]he claims are rejected as falling under the judicial exception of an abstract idea … not statutory within the meaning of 35 U.S.C. 101” (Ans. 3), “an applicant may show that a process claim satisfies § 101 either by showing that his claim is tied to a particular machine, or by showing that his claim transforms an article” into a different state or thing. *See Gottschalk v. Benson*, 409 U.S. 63, 70 (1972).

Claim 1 recites a “computer implemented method” to compare “a first object and a second object in an object-oriented operating system.” We find that comparing objects on a computer is a method that requires performance on a computer. Thus, we conclude that claim 1 is “tied to” a machine, i.e., a

computer. Accordingly, we agree with Appellant that the computer-implemented method of claim 1 satisfies 35 U.S.C. § 101.

Though claim 1 requires an algorithm of determining whether the objects are equal, the claim also requires obtaining methods from the objects to determine whether the methods are equal, generating a document listing the differences between the methods, and then transforming the document into human-readable form to be displayed. The claimed method, if allowed, would not patent an abstract idea or pre-empt any other uses of the mathematical formula, but rather give a limited right to exclude of the claimed computer-implemented method to the Appellant.

For the foregoing reasons, we cannot sustain the rejection of claim 1 and claims 3-5 and 7-9 standing therewith under 35 U.S.C. § 101.

35 U.S.C. § 103(a)

Appellant contends that “*Mason* describes a system for testing various parameters of Java Web applications to determine if the applications succeed” (App. Br. 7), and thus Mason does not disclose “‘determining whether the first object is equal to the second object; and if said objects are not equal,’ or ‘obtaining one or more methods from said first object and said second object’” (*id.*). Appellant further contends that the sections of Mason cited by the Examiner “has nothing to do with testing the equality of methods of two objects to be tested” (App. Br. 8).

Though the Examiner finds that “[b]oth the invention of Mason and the claimed invention operate on the Java operating system” and that “both inventions deal with the same problem (comparing two objects) and arrive at the same solution” (Ans. 9-10), we disagree with the Examiner that Mason discloses “determining whether the first object is equal to the second object”

and “if said objects are not equal . . . obtaining one or more methods from said first object and said second object” as required by claim 1.

In particular, Mason discloses testing applications (FF 1), wherein a sub-elements of EJB module is compared another sub-element value of a quality of service (QOS) element (FF 2). Though we find that the sub-element values may comprise methods of an object, and thus agree with the Examiner that Mason at least suggests “determining whether the one or more methods from said first object are equal to the one or more methods from said second object” as required in claim 1, we agree with Appellant that Mason does not disclose ““determining whether the first object is equal to the second object; and if said objects are not equal,’ or ‘obtaining one or more methods from said first object and said second object”” (App. Br. 7). That is, even if Mason’s sub-element values that are compared are the methods of a first object and a second object as required by claim 1, such first object and the second object of Mason were not determined to be equal before the methods were obtained from the objects. We find that the sections of Mason referenced by the Examiner are silent as to any determination of whether the objects are equal in order to obtain the methods, as required by claim 1.

Java and Keller do not cure the deficiencies of Mason. The Examiner has not shown, and we do not find, that the combination of Mason, Java and Keller teaches or would have reasonably suggested the argued limitations. We, therefore, conclude the positions of the Examiner are untenable.

Because we agree with at least one of the Appellant’s contentions, we

Appeal 2009-002724
Application 10/667,816

cannot sustain the obviousness rejection of claim 1 and claims 3-5 and 7-9 standing therewith. Accordingly, we cannot sustain the obviousness rejection of claims 1, 3-5 and 7-9.

V. CONCLUSION

We conclude that the Appellant has shown that the Examiner erred in determining the method recited in claims 1, 3-5, and 7-9 is not patent-eligible subject matter under 35 U.S.C. § 101.

We also conclude that the Appellant has shown that the Examiner erred in finding that the combination of Mason in view of Java and Keller would have suggested “determining whether the first object is equal to the second object” and “if said objects are not equal... obtaining one or more methods from said first object and said second object” as required by claim 1.

VI. DECISION

We have not sustained the Examiner's rejection with respect to any claim on appeal. Therefore, the Examiner's decision rejecting claims 1, 3-5, and 7-9 is reversed.

REVERSED

peb

MICHAEL B. JOHANNESEN, ESQ.
LOWENSTEIN SANDLER, P.C.
65 LIVINGSTON AVENUE
ROSELAND, NJ 07068